

BHYWI-08: Semester-Fahrplan 2019

Vorlesungen

Datum	V	Thema
05.04.2019	01	IT: GitHub / Qt Installation
12.04.2019	02	Grundlagen: Kontinuumsmechanik
19.04.2019	--	Ostern
26.04.2019	03	Grundlagen: Hydromechanik
03.05.2019	04	Grundlagen: Partielle Differentialgleichungen
10.05.2019	05	Grundlagen: Numerik, Qt Übung: Funktionsrechner
17.05.2019	06	Einführung in Python
24.05.2019	07	Numerik: Finite Differenzen Methode I (explizit)
31.05.2019	08	Hausaufgabe
07.06.2019	09	Numerik: Finite Differenzen Methode II (implizit)
14.06.2019		Pfingsten
21.06.2019	10	Gerinnehydraulik: Theorie – Grundlagen
28.06.2019	11	Gerinnehydraulik: Programmierung, Übung
05.07.2019	12	Grundwassermodellierung: Datenbasierte Methoden
12.07.2019	13	Beleg

Hydroinformatik II

”Prozesssimulation und Systemanalyse”

BHYWI-08-07 @ 2019

Finite-Differenzen-Methode II

Olaf Kolditz

*Helmholtz Centre for Environmental Research – UFZ

¹Technische Universität Dresden – TUDD

²Centre for Advanced Water Research – CAWR

07.06.2019 - Dresden

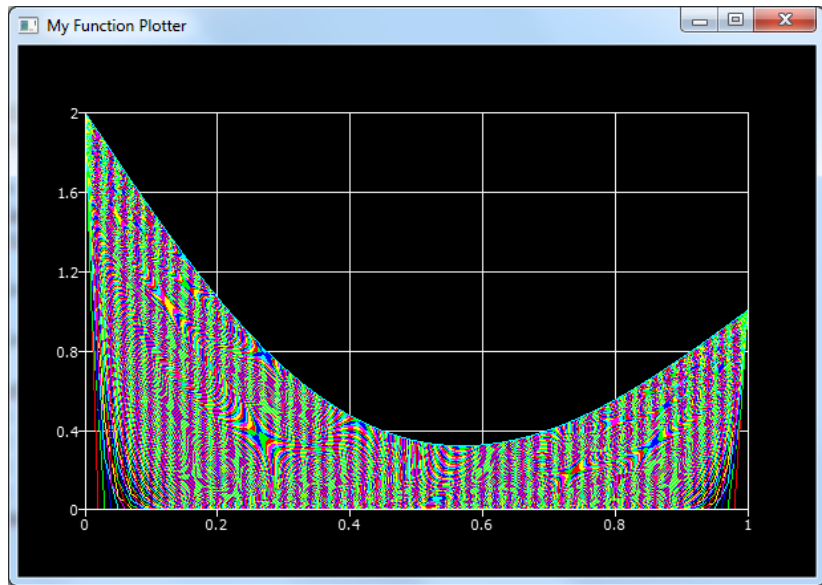
1 BHYWI-08-03: Übung explizite FDM

2 L08-07: Implicit FDM for diffusion equation

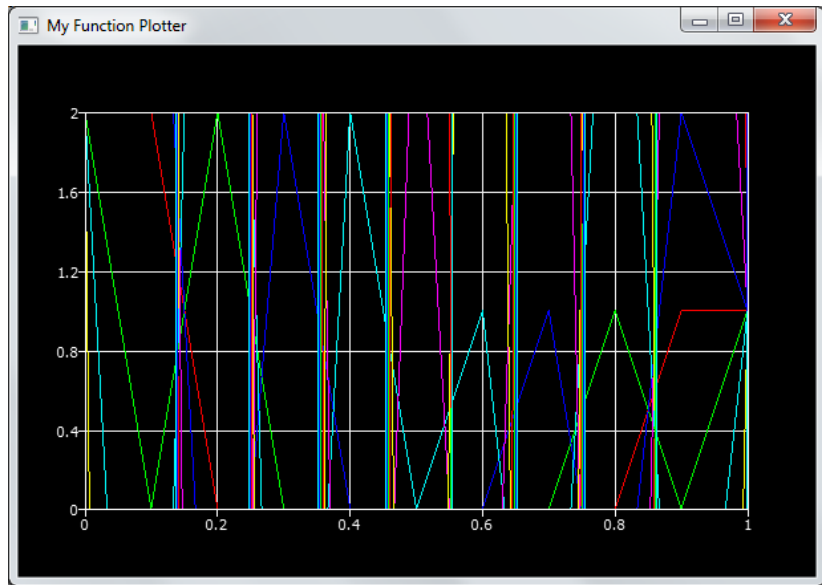
3 BHYWI-08-04: Übung implizite FDM

4 Qt Basics

Letzte Vorlesung: explizite FDM

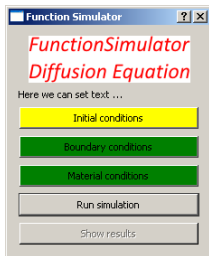
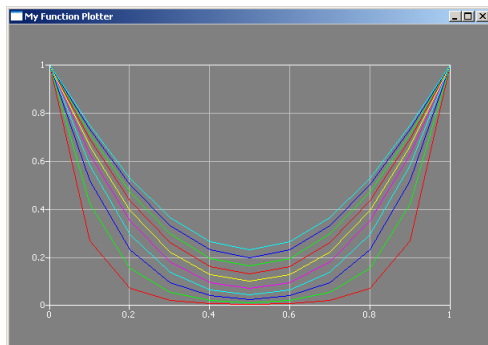
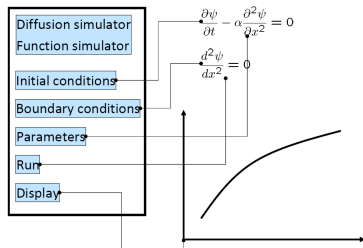


Letzte Vorlesung: explizite FDM - Zeitschrittbegrenzung



$$Ne = \alpha \frac{\Delta t}{\Delta x^2} \leq 0.5 \quad (1)$$

Ziel der Vorlesung



- ▶ PDE for diffusion processes

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0 \quad (2)$$

- ▶ Time discretization

$$\left[\frac{\partial u}{\partial t} \right]_j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t} \quad (3)$$

- ▶ Forward time / centered space

$$\left[\frac{\partial^2 u}{\partial x^2} \right]_j^{n+1} \approx \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} \quad (4)$$

- ▶ (Current time / centered space)

$$\left[\frac{\partial^2 u}{\partial x^2} \right]_j^n \approx \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2}$$

- ▶ Substitute into PDE

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} - \alpha \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} = 0 \quad (6)$$

- ▶ Algebraic equation (index notation)

$$\frac{\alpha \Delta t}{\Delta x^2} (-u_{j-1}^{n+1} + 2u_j^{n+1} - u_{j+1}^{n+1}) + u_j^{n+1} = u_j^n \quad (7)$$

- ▶ Algebraic equation (matrix notation)

$$\mathbf{Ax} = \mathbf{b} \quad (8)$$

- ▶ Explain steps with black board


- ▶ Algebraic equation (matrix notation)

$$\mathbf{Ax} = \mathbf{b} \quad (9)$$

- ▶ Algebraic equation (index notation)

$$Ne (-u_{j-1}^{n+1} + 2u_j^{n+1} - u_{j+1}^{n+1}) + u_j^{n+1} = u_j^n \quad (10)$$

- ▶ Let's take a closer look ...

$$\underbrace{\begin{bmatrix} 1 + 2Ne & -Ne & & & & & \\ -Ne & \dots & \dots & & & & \\ & \dots & \dots & \dots & & & \\ & & \dots & \dots & -Ne & & \\ & & & -Ne & 1 + 2Ne & & \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \dots \\ u_{n-1} \\ u_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_{n-1} \\ b_n \end{bmatrix}}_{\mathbf{b}} \quad (11)$$


► Data structures (as usual ...)

```
Dialog::Dialog(QWidget *parent) : QDialog(parent)
{
    matrix = new double[n*n];
    vecb = new double[n];
    vecx = new double[n];
}
```

```
Dialog::~Dialog()
{
    delete [] matrix;
    delete [] vecb;
    delete [] vecx;
}
```

► Functions (a pain in the neck ...)

```
AssembleEquationSystem();  
Gauss(matrix, vecb, vecx, n);
```

```
void Dialog::AssembleEquationSystem()  
{...  
    int i,j;  
    // Matrix entries  
    for(i=0;i<n;i++)  
    {  
        vecb[i] = u_old[i]; // RHS Vektor  
        for(j=0;j<n;j++)  
        {  
            matrix[i*n+j] = 0.0;  
            if(i==j) // Hauptdiagonale  
                matrix[i*n+j] = 1. + 2.*Ne;  
            else if(abs((i-j))==1) // Nebendiagonalen  
                matrix[i*n+j] = - Ne;  
        }  
    }  
    ...}
```

- ▶ Boundary conditions - concept

$$\mathbf{Ax} = \mathbf{b} \quad (12)$$

$$[1 \ 0 \ 0 \ \dots \ 0] \begin{bmatrix} u_0 \\ u_1 \\ \dots \\ u_n \end{bmatrix} = \begin{bmatrix} u_0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (13)$$

► Boundary conditions - implementation

```
void Dialog::AssembleEquationSystem()
{...
  // Treat boundary conditions
  for(i=0;i<n;i++)
    for(j=0;j<n;j++)
      {
        if(i==0||i==n-1)
          matrix[i*n+j] = 0.0;
      }
  for(i=0;i<n;i++)
    {
      if(i!=0&& i!=n-1)
        continue;
      for(j=0;j<n;j++)
        {
          if(i==j)
            matrix[i*n+j] = 1.0;
          else
            matrix[i*n+j] = 0.0;
        }
    }
}
```

Solving EQS - How to ... the magic Gauss function

▶ 1

$$a_{11}u_1 + a_{12}u_2 = b_1 \quad (14)$$

$$a_{21}u_1 + a_{22}u_2 = b_2 \quad (15)$$

▶ 2

$$a_{21} \frac{a_{11}}{a_{21}} u_1 + a_{22} \frac{a_{11}}{a_{21}} u_2 = \frac{a_{11}}{a_{21}} b_2 \quad (16)$$

▶ 3

$$\left(\frac{a_{22}a_{11}}{a_{21}} - a_{12} \right) u_2 = \frac{a_{11}}{a_{21}} b_2 - b_1 \quad (17)$$

▶ 4

$$u_2 = \frac{\frac{a_{11}}{a_{21}} b_2 - b_1}{\frac{a_{22}a_{11}}{a_{21}} - a_{12}} \quad (18)$$

Implementation #5

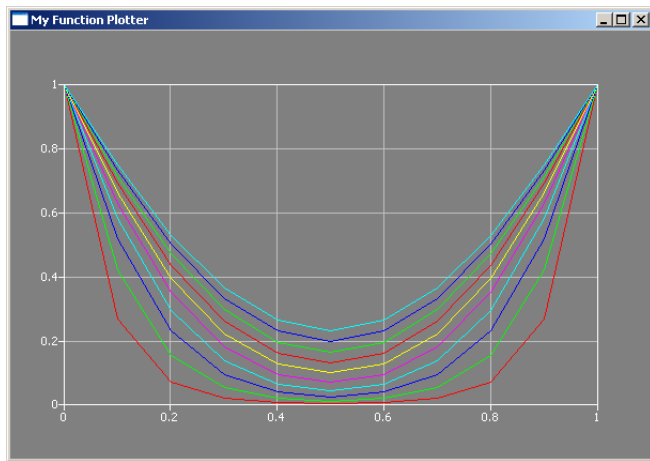
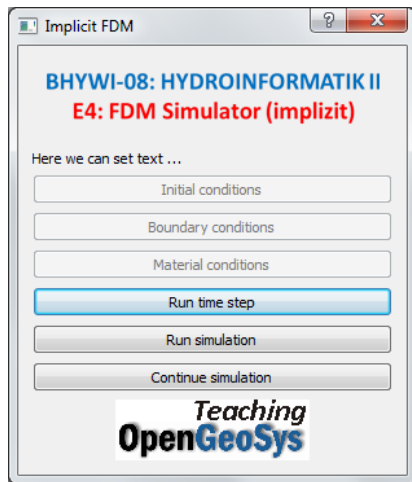


Figure: Zeitliche Entwicklung des Diffusionsprofils - implizites Verfahren (Wahoo...)

Implementation #6: Run multiple time steps



- Einzelne Zeitschritte
- Mehrere Zeitschritte
- Weiterführen der Berechnung

```
void RunTimeStep();  
void RunTimeLoop();  
void ContinueTimeLoop();
```

- ▶ ... just a star (*)

```
void Dialog::on_pushButtonSH0_clicked()
{
    Plotter *plotter = new Plotter;
    ...
    plotter->show();
}
```

- ▶ ... for better plotting (in your life)

```
void Dialog::SH0Better()
{
    ...
    plotterAIO->show();
}
```

C++Basics (... the difference (the whole story))

- ▶ Plotter declaration

```
class Dialog : public QDialog
{...
private:
    Plotter *plotterAIO;
}
```

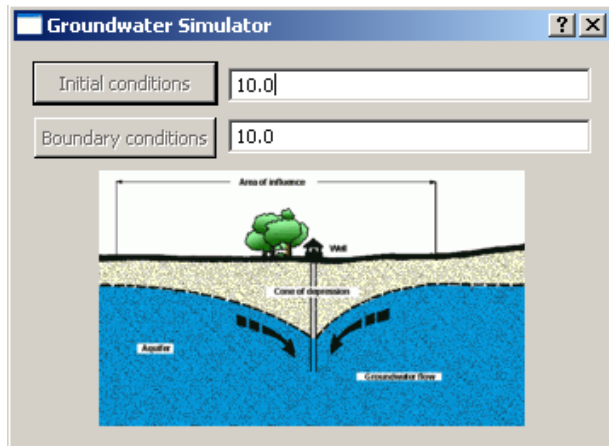
- ▶ Plotter declaration ... otherwise

```
Dialog::Dialog(QWidget *parent) : QDialog(parent)
{...
    plotterAIO = new Plotter;
}
```

- ▶ Finally ready to use ...

```
void Dialog::SHOBetter()
{...
    plotterAIO->show();
}
```

► QLineEdit



Datum	E	Übungen
05.04.2019	00	Git und QT (Lars Bilke)
03.05.2019	01	Qt: Hallo World
10.05.2019	02	Qt: Funktionsrechner
17.05.2019		
	03	Qt: Explizite Finite-Differenzen-Methode
	04	Qt: Implizite Finite-Differenzen-Methode
	05	Qt: Gerinnehydraulik I (QAD)
	06	Qt: Gerinnehydraulik II (OOP)
	08	Qt: Gerinnehydraulik IV (interaktiv)
		...

<https://github.com/envinf/Hydroinformatik-II>